TITLE:

An Improvement of the Soundness of a 3-bit PCP (Theoretical Computer Science and Its Applications)

AUTHOR(S):

Kinoshita, Naoki; Tamaki, Suguru; Iwama, Kazuo

CITATION:

Kinoshita, Naoki ...[et al]. An Improvement of the Soundness of a 3-bit PCP (Theoretical Computer Science and Its Applications). 数理解析研究所講究録 2009, 1649: 129-136

ISSUE DATE:

2009-05

URL:

http://hdl.handle.net/2433/140746

RIGHT:

京都大学学術情報リポジトリ
KURENAI 紅
Kyoto University Research Information Repository

数理解析研究所講究録
第1649巻 2009年 129-136

An Improvement of the Soundness of a 3-bit PCP

Naoki Kinoshita*
kinoshita@kuis.kyoto-u.ac.jp

Suguru Tamaki*
tamak@kuis.kyoto-u.ac.jp

Kazuo Iwama*
iwama@kuis.kyoto-u.ac.jp

Abstract

Approximation algorithms have been studied to cope with computationally hard combinatorial problems such as NP-hard problems, for which we cannot hope for exact solutions efficiently. Approximation algorithms compute feasible solutions with some theoretically guaranteed quality in polynomial time. Probabilistically Checkable Proofs (PCPs) have been succeeded to show the limitation of such approximation algorithms, that is, how good approximate solutions can be compared to optimal solution.

PCP is a mathematical model which probabilistically recognizes a certain (especially NP) language L by making queries to a kind of oracle called a proof. There are some important parameters which characterize PCPs. Completeness (soundness) is the maximum probability that a PCP accepts an input which is in (respectively, not in) L. The number of queries to the proof and the adaptivity in queries, that means a dependency between the queries, are also important aspects. In this paper, we study how small the soundness of a PCP can be when it has perfect completeness and makes non-adaptive three queries.

We can show better hardness of approximation of the optimization problem corresponding to a PCP if we can construct a PCP with smaller soundness. Khot and Saket obtained a PCP with soundness value $\frac{20}{27}+\epsilon \simeq 0.74074$, that probabilistically selects one of four tests and perform it to the proof. We show that the soundness can be $\frac{16+\sqrt{6}}{25}+\epsilon \simeq 0.73798$ by optimizing the probability of selecting each test. Here ϵ is an arbitrarily small constant. As a result of our optimization, one of the four tests are shown to be unnecessary.

1 Introduction

Most natural optimization problems arising in application areas are NP-hard. It is widely believed that we need super-polynomial time to obtain optimal solutions for them. Thus, from the perspective of practice, it is important to consider approximation algorithms for obtaining approximate solutions that may not the same as but close to optimal solutions. In approximation algorithms, to guarantee the ratio of the quality of an approximate solution to that of an optimal solution is very important issue. The proximity of an approximate solution to an optimal solution is defined as following:

$$\text{approximation ratio} = \min \frac{\text{quality of approximate solution}}{\text{quality of optimal solution}}$$

Here we consider maximization problems and assume that the value of the quality of a solution is positive, and min is taken over all the possible inputs for the algorithm. Hence the approximation ratio is a value between 0 to 1 and larger value means better. The best possible approximation ratio of polynomial time algorithms differs according to each problem. Giving upper bounds on an approximation ratio of certain problem is an important theme in the study of approximation algorithms and is called the hardness of approximation or inapproximability.

A typical way to prove the hardness of approximation is the reduction from PCP. We consider a PCP that has perfect completeness, that means the completeness is 1, and makes three non-adaptive queries to the proof. Perfect completeness means that all the inputs which should be accepted are accepted with proper proofs. Non-adaptive means the content of every query is independent from any other queries.

We describe some previous results around 3-bit PCPs below. If we are allowed to loose perfect completeness, there is a well-known PCP due to Håstad [6], that has completeness $1-\epsilon$, soundness $\frac{1}{2}+\epsilon$ and makes non-adaptive queries, Here, ϵ is an arbitrarily small constant. When assuming perfect completeness, Guruswami et al. [5] showed a PCP with soundness $\frac{1}{2}+\epsilon$ and adaptive queries. Above both results are tight, that is, the PCP with smaller soundness always looses the ability of recognizing NP languages. Therefore, there is a trade-off between non-adaptiveness and perfect completeness of a PCP. For the PCP of this paper's scope, Khot and Saket [10] showed the current smallest soundness value $\frac{20}{27}+\epsilon$ with perfect completeness and three non-adaptive queries. In this paper, we show that the soundness can be improved to $\frac{16+\sqrt{6}}{25}+\epsilon$. There may be still a room to improve the soundness as Zwick [13] conjectured that the soundness can be possibly improved to $\frac{5}{8}$. Recently, O'Donnell and Wu [11] showed that Zwick's conjecture is true, i.e., the soundness $\frac{5}{8}$ can be achieved under Khot's d-to-1 Conjecture [8] which is a stronger than standard P$\neq$NP assumption.

*Graduate School of Informatics, Kyoto University

2 Probabilistically Checkable Proofs

2.1 Definitions

A language class recognized by a PCP is defined as follows.

Definition 1. (Probabilistically Checkable Proof) *A language L is in a language class $\mathrm{PCP}_{c,s}[r(n), q(n)]$ if there exists a polynomial time probabilistic Turing machine V s.t. given an input x of size n and a proof Π, V*

- *Uses $r(n)$ random bits and makes $q(n)$ bits of queries to Π.*
- *Accepts or rejects x according to the result of the queries.*
- *has these two properties:*
  - **Completeness:** *If $x \in L$, there exists a proof Π such that V accepts x with probability at least c.*
  - **Soundness:** *If $x \notin L$, for any proof Π, x is accepted by V with probability at most s.*

Parameters c, s are called completeness and soundness respectively. To clarify whether the queries are done adaptively or non-adaptively, we use the notation $\mathrm{aPCP}, \mathrm{naPCP}$. The PCP is said to have perfect completeness if its completeness is 1. That is a desirable property since we can reduce the soundness of such PCPs to arbitrarily small constant by repeating verification. In this study we only consider PCPs with perfect completeness. As for the power of PCPs, the following PCP Theorem is well known.

Theorem 2. (PCP Theorem, Arora et al.[1, 2]) $\mathrm{NP} = \mathrm{naPCP}_{1,\frac{1}{2}}[O(\log n), O(1)]$

This theorem means that for an arbitrary NP language L, there is a PCP such that if a string x is in L, then it can be always accepted by the PCP by giving a proper proof, and if x is not in L, then it cannot be accepted by the PCP with probability more than $\frac{1}{2}$ by using any proof. Furthermore, it shows that such PCPs need only constant number of queries to the proofs regardless of input length. Intuitively, reducing the number of queries in PCP with keeping the same soundness implies better inapproximability results.

2.2 2 Prover 1 Round Games

The process of a PCP can be viewed as a game between the prover who gives a proof and the verifier who verifies the input by questioning to the prover. In other words, the prover tries to make verifier accept regardless of the input and the verifier tries to accept only the correct input by detecting prover's lies. We can think of PCP proofs as two dynamic provers P_1 and P_2 rather than a bit string. This interpretation often makes the design of better PCPs easier. In this case, the provers answer to the verifier's question obey their pre-determined strategies A and B respectively. They are not allowed to communicate each other nor know the question to another prover after the PCP verification process starts. We need such restrictions to assure that the verification ability of the entire system does not increase since the proof changes from a static string to two dynamic provers. Such games are called 2-prover 1-round games (2P1R games). We define the value of a game G as follows.

Definition 3. (The value of a game G) *Given a game G, denote by $V(x) = \mathrm{Acc}$ an event that the verifier V accepts the input x according to the strategies A and B of provers P_1 and P_2. Then the value of the game G, denoted by $\omega(G)$, is defined as*

$$\omega(G) = \max_{A,B} \Pr[V(x) = \mathrm{Acc}].$$

2.3 Parallel Repetition

The ratio of soundness and completeness of a PCP is closely related to the approximability of a certain optimization problem. It gives upper bounds on the approximation ratio achieved by polynomial time algorithms for the problem. Thus, a PCP with smaller soundness is desired to obtain better inapproximability results. Soundness represents the probability that the PCP accepts an input which must be rejected. Let us think of a PCP as a 2P1R game G and consider a new game where we perform the same game independently for u times to the same input and accept if the input is accepted by all games. In this case, the value of the entire game becomes $\omega(G)^u$, and if the completeness is 1 (or sufficiently close to 1), then the soundness can be made exponentially (and thus arbitrarily) small w.r.t. u. However, notice that if we repeat the game independently, the game becomes u-round game and the relation to the PCP does not hold any longer. To keep the correspondence to PCP, we pose a restriction that the verifier should send questions of u times repetition at once. We write such a 1-round game G^u and call a parallel repetition of a game. The value of the game G^u does not decrease as $\omega(G)^u$ in case of independently repeated games, but it also decreases exponentially w.r.t. u, as shown by Raz.

Theorem 4. (Parallel Repetition Theorem, Raz [12]) *Given a 2P1R game G with soundness $s < 1$ and answer set of size d, there exists a constant $s' < 1$ which depends only on s, s.t. G's u times parallel repetition G^u has the soundness $(s')^{\frac{u}{d}}$.*

Although we can decrease the soundness by parallel repetition, there is a drawback, that is, the number of query bits also becomes u times as the original game. Thus it cannot be used in a straightforward way to improve the soundness of the PCP. To construct PCPs with small number of query bits, we can use a technique of "composition"; we can combine this parallel repeated games with arbitrarily small soundness, called outer verifier, and another verifier which uses small number of query bits, called inner verifier, to make soundness and number of queries small at the same time.

2.4 Long Code

Although an outer verifier can have large number of query bits, an inner verifier is required to significantly reduce the number of query bits. To do so, encoding a proof in some proper way is very helpful. Currently the most useful encoding way is the one introduced by Bellare et al. [3], called a Long Code.

Definition 5. (Long Code) *Let $\mathcal{F}_{\mathcal{M}} = \{f \mid f : \mathcal{M} \mapsto \{0,1\}\}$. The long code of an element $x \in \mathcal{M}$ is defined by a map as*

$$A_x : \mathcal{F}_{\mathcal{M}} \mapsto \{0,1\}, \quad A_x(f) = f(x).$$

Given an n bit input, Let G be a corresponding 2P1R game where the verifier V makes total T bits of queries to P_1 and P_2 and consider the parallel repetition G^u. In G^u, the number of the queries made by the verifier becomes Tu. We would like to reduce it to some constant number independent of u by encoding the provers' strategies in some way. Let the set of answers in G^u be $\mathcal{M} = \{0,1\}^{Tu}$. V would like to decide whether the answer $x \in \mathcal{M}$ from the provers is in the subset $S \subseteq \mathcal{M}$ of answers which satisfy the condition of acceptance. Now, define a function f_S as

$$f_S : \mathcal{M} \mapsto \{0,1\}, \quad f_S(x) = \begin{cases} 1 & (x \in S) \\ 0 & (x \notin S) \end{cases}.$$

Then, we can check whether $x \in S$ or not by only 1 bit query of $A_x(f_S)$. This gives an intuitive explanation for reducing query bits by Long Code. Of course, the above 1 bit verification can be easily cheated by provers P_1 and P_2's strategies of setting $\forall f, A(f) = 1$. Thus V is also required to check whether the strategy A of the provers satisfies the definition of a Long Code, that is, $\exists x, \forall f, A(f) = f(x)$. We can perform such a verification by using 3-bits, which is the known smallest number for it, and we will focus on such 3-bits tests. Note that the size of a Long Code A_x to some $x \in \mathcal{M}$ is calculated as

$$|A_x| = |\mathcal{F}_{\mathcal{M}}| = 2^{|\mathcal{M}|} = 2^{2^{Tu}}.$$

It means that in G^u, the domain size of the queries asked by V is represented by $2^{2^{Tu}}$ bits. Since both T and u are constants that do not depend on input size n, the size $2^{2^{Tu}}$ is also constant. Thus, in G^u, V can perform queries to P_1 and P_2 in a constant time.

3 3-bit PCP

We focus on non-adaptive 3-bit PCPs with perfect completeness. The construction due to Khot and Saket achieves the current smallest soundness value, as presented in the rest of this section.

Theorem 6. (Khot et al. [10])

$$\forall \epsilon > 0, \quad \mathrm{NP} = \mathrm{naPCP}_{1,\frac{29}{27}+\epsilon}[O(\log n), 3]$$

In this section, we look inside their PCP construction.

3.1 Outer Verifier

The outer verifier used in Khot-Saket's PCP is a tester for satisfiability of a 3-SAT formula and its soundness is reduced by parallel repetition. The following theorem about the complexity of 3SAT is known.

Theorem 7. ([9]) *There exists an universal constant $c < 1$ such that distinguishing that a 3-SAT formula ψ is satisfiable (YES instance) or no more than c fraction of clauses can be satisfied simultaneously by any assignment is NP-hard. Moreover, the above statement holds even when formulas have canonical property that every clause consists of exactly three literals and every variable appears exactly five times in them. We call a 3-SAT formula with the above properties a 3-SAT-5 instance.*

The 2P1R game used here is constructed by Khot [7]. We denote by V_{2P1R} the outer verifier explained here to distinguish it from the inner verifier which is of our interest for improvement.

First, let $\{x_1, x_2, \ldots\}$ be Boolean variables and $\{C_1, C_2, \ldots\}$ be the clauses in a 3-SAT-5 instance ψ. This game is parametrized by T and u. We assume $T, u \gg 1$ and these parameters can take arbitrarily large values independently. The verifier V_{2P1R} chooses Tu random clauses from ψ and let these clauses be $W = \{C_1, C_2, \ldots, C_{Tu}\}$. Let W a question to P_1 and P_1 answers an assignment to the clauses in W needed for ϕ to be satisfied. We describe the set of possible satisfied assignments to W by $\mathcal{M}_W$ and an answer of P_1 by $\sigma \in \mathcal{M}_W$. Next V_{2P1R} chooses a subset of W of size u at random. We describe the subset by $S = \{C_{i_1}, C_{i_2}, \ldots, C_{i_u}\}$ and assume the order as $1 \le i_1 < i_2 < \cdots < i_u \le Tu$. Each clause C_{i_j} contains three variable and V_{2P1R} chooses from each clause a variable x_{i_j}. Let $U = \{x_{i_1}, x_{i_2}, \ldots, x_{i_u}\} \cup (W \backslash S)$. Note that U is a family of u variables and $(T-1)u$ clauses. U is a question to the prover P_2 and P_2 answers an assignment to the variables and clauses in U needed for ϕ to be satisfied. We describe the set of these assignment by $\mathcal{M}_U$ and an answer of P_2 by $\tau \in \mathcal{M}_U$. Any assignment to W can be restricted to an assignment to U and confirming such relation is the consistency test of V_{2P1R}. Mapping $\pi^{W,U} : \mathcal{M}_W \mapsto \mathcal{M}_U$ represents the restriction from an assignment W to U and V_{2P1R} accepts iff $\pi^{W,U}(\sigma) = \tau$.

Here we describe the set of all possible questions to P_1 by $\mathcal{W}$ and to P_2 by $\mathcal{U}$. Obviously, if the formula ψ is an YES instance (satisfiable), P_1 and P_2 have a strategy which makes V_{2P1R} accept with probability 1. The strategy is that first fix a satisfiable assignment to ψ and both provers answer the exact assignment.

If ψ is a NO instance (no more than c fraction of clauses can be satisfied by any assignment), by theorem 4, the following holds.

Theorem 8. *If ψ is NO instance, with any strategy, P_1 and P_2 cannot make V_{2P1R} accept with probability higher than c_0^u. Here c_0 is an universal constant.*

3.2 Biased Long Code

We review biased Long Code and its Fourier transformation used in the analysis of Khot-Saket's PCP. In what follows, we represent Boolean value of $\{\text{true}, \text{false}\}$ by $\{-1, 1\}$ instead of $\{1, 0\}$. With such a notation, we can write exclusive-or of Boolean variables by arithmetic multiplication.

Biased long code has bias on the distribution of its index function $f \in \mathcal{F}_{\mathcal{M}}$. Here, $\mathcal{F}_{\mathcal{M}} = \{f \mid f : \mathcal{M} \mapsto \{-1, 1\}\}$. Let the bias $0 < p < 1$ and the function f is selected for every $x \in \mathcal{M}$ to be $f(x) = -1$ with probability p and $f(x) = 1$ with probability $1 - p$. We denote this as $f \in_R \mu_p(\mathcal{M})$.

Fourier transformation is used for the analysis of tests with long code. Biased case of Fourier analysis is known for example in [4] and we need to choose appropriate orthonormal basis. The space of all tables $A : \mathcal{F}_{\mathcal{M}} \mapsto \mathbf{R}$ forms a $2^{|\mathcal{M}|}$ dimensional real vector space and addition of two tables A_1 and A_2 is defined at each point as

$$(A_1 + A_2)(f) = A_1(f) + A_2(f).$$

Inner product in this space is defined as

$$\langle A_1, A_2 \rangle = E_{f \in_R \mu_p(\mathcal{M})}[A_1(f) A_2(f)].$$

For every $x \in \mathcal{M}$, a function $\phi_x : \mathcal{F}_{\mathcal{M}} \mapsto \mathbf{R}$ is defined as

$$\phi_x(f) = \begin{cases} -\sqrt{q/p} & \text{if } f(x) = -1 \\ \sqrt{p/q} & \text{if } f(x) = 1 \end{cases}.$$

The orthonormal basis is represented as below. For every subset $\beta \subseteq \mathcal{M}$, define $\chi_\beta : \mathcal{F}_{\mathcal{M}} \mapsto \mathbf{R}$ as

$$\chi_\beta = \prod_{x \in \beta} \phi_x.$$

In above definition, $\chi_\emptyset = 1$ holds and for any $x \in \mathcal{M}$, $\chi_{\{x\}} = \phi_x$ holds. Hence every table can be represented as

$$A = \sum_{\beta \subseteq \mathcal{M}} \hat{A}_\beta \chi_\beta.$$

Here $\hat{A}_\beta$ is called a Fourier coefficients, which is calculated as

$$\hat{A}_\beta = \langle A, \chi_\beta \rangle.$$

When the range of A is $\{-1, 1\}$, $\sum_\beta \hat{A}_\beta^2 = 1$ holds by Parseval's identity.

3.3 Test of the Verifier

The PCP consists of four tests. In this section, we describe the behavior and some part of the analysis of the tests. Each test is performed with probability defined later. We define the bias on queries by $p = \frac{1}{2} + \epsilon, q = 1 - p$ and parameter $\epsilon > 0$ is assumed to be taken arbitrarily small. Also, let $T = \frac{1}{\epsilon^4}$ and u is taken sufficiently large.

3.3.1 Test T_1

Test T_1 checks the consistency between long codes A and B. The procedure of the test T_1 is following.

1. According to the method of the verifier $V_{2\text{P1R}}$, choose random set $W \in \mathcal{W}$ and its subset $U \in \mathcal{U}$. The verifier expect $B : \mathcal{F}_{\mathcal{M}_W} \mapsto \{-1, 1\}$ and $A : \mathcal{F}_{\mathcal{M}_U} \mapsto \{-1, 1\}$ to be the long codes corresponding to the assignments to W and U respectively. Let $\pi = \pi^{W,U}$ the mapping from W to U.
2. Choose functions $f \in_R \mu_p(\mathcal{M}_U)$ and $g \in_R \mu_p(\mathcal{M}_W)$ independently.
3. For each $y \in \mathcal{M}_W$, choose a function $h : \mathcal{M}_W \mapsto \{-1, 1\}$ as
   - $f(\pi(y)) = 1 \wedge g(y) = 1 \to h(y) = 1$
   - $f(\pi(y)) = 1 \wedge g(y) = -1 \to h(y) = -1$
   - $f(\pi(y)) = -1 \wedge g(y) = 1 \to h(y) = -1$
   - $f(\pi(y)) = -1 \wedge g(y) = -1 \to$
     $h(y) = \begin{cases} 1 & \text{with prob. } q/p \\ -1 & \text{with prob. } 1 - q/p \end{cases}$
4. Accept iff $(A(f), B(g), B(h)) \in S_1$. Where $S_1 = \{(1,1,1), (1,-1,-1), (-1,1,-1), (-1,-1,1), (-1,-1,-1)\}$

Let us consider the completeness of the test T_1. When the 3-SAT-5 formula ψ is an YES instance, there exists a strategy which makes the outer verifier $V_{2\text{P1R}}$ always accept. That is, fix one of the satisfiable assignment and answer the queries of the verifier according to the assignment. Pick some parts $x \in \mathcal{M}_U, y \in \mathcal{M}_W, \pi(y) = x$ of the satisfiable assignment corresponding to U and W and let them long codes A and B. Due to the definition of the long code, $(A(f), B(g), B(h)) = (f(x), g(y), h(y)) = (f(\pi(y)), g(y), h(y))$ holds and now (f, g, h) are chosen always as

$$\forall y \in \mathcal{M}_W, \quad (f(\pi(y)), g(y), g(h)) \in S_1,$$

so the PCP verifier always accept. Thus, T_1 has perfect completeness.

Next, to consider the soundness of the test T_1, assume the formula ψ is a NO instance. We bound the acceptance probability of the verifier using Fourier analysis. Here the following holds.

Lemma 9. ([10]) *When $x, y, z \in \{-1, 1\}$, expression*

$$\frac{5 - x - y - z + xy + xz + yz + 3xyz}{8}$$

has value 1 only when $(x, y, z) \in S_1$ and 0 otherwise.

Using above lemma, the acceptance probability of the verifier can be represented by

$$\begin{aligned} \Pr[\text{Acc}] = E_{W,U,f,g,h}\Big[\frac{1}{8}\big(5 &- A(f) - B(g) - B(h) \\ &+ A(f)B(g) + A(f)B(h) \\ &+ B(g)B(h) + 3A(f)B(g)B(h)\big)\Big]. \end{aligned}$$

We use Fourier transform on the above expression and bound all terms, then we have

$$\Pr[\mathrm{Acc}] \leq E_{W,U}\left[\frac{5-\hat{A}_\emptyset-2\hat{B}_\emptyset+2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2+3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8}\right]+O(\epsilon).$$

3.3.2 Test T_2

Test T_2 is a slight modification of test T_1 where the choice of the function h and the acceptance condition is different. The procedure of the test T_2 is following.

1. According to the method of the verifier $V_{2\mathrm{P1R}}$, choose random set $W \in \mathcal{W}$ and its subset $U \in \mathcal{U}$. The verifier expect $B : \mathcal{F}_{\mathcal{M}_W} \mapsto \{-1,1\}$ and $A : \mathcal{F}_{\mathcal{M}_U} \mapsto \{-1,1\}$ to be the long codes corresponding to the assignments to W and U respectively. Let $\pi = \pi^{W,U}$ the mapping from W to U.

2. Choose functions $f \in_R \mu_p(\mathcal{M}_U)$ and $g \in_R \mu_p(\mathcal{M}_W)$ independently.

3. For each $y \in \mathcal{M}_W$, choose a function $h : \mathcal{M}_W \mapsto \{-1,1\}$ as

   - $f(\pi(y)) = 1 \wedge g(y) = 1 \rightarrow h(y) = -1$
   - $f(\pi(y)) = 1 \wedge g(y) = -1 \rightarrow$ $h(y) = \begin{cases} 1 & \text{with prob. } q/p \\ -1 & \text{with prob. } 1-q/p \end{cases}$
   - $f(\pi(y)) = -1 \wedge g(y) = 1 \rightarrow h(y) = 1$
   - $f(\pi(y)) = -1 \wedge g(y) = -1 \rightarrow h(y) = -1$

4. Accept iff $(A(f), B(g), B(h)) \in S_1$. Where $S_2 = \{(1,1,-1),(1,-1,1),(1,-1,-1),$ $(-1,1,1),(-1,-1,-1)\}$

We can check that test T_2 has perfect completeness in the same way as T_1. And the acceptance probability for NO instance satisfies following expression.

$$\Pr[\mathrm{Acc}] \leq E_{W,U}\left[\frac{5+\hat{A}_\emptyset-2\hat{B}_\emptyset-2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2-3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8}\right]+O(\epsilon)$$

3.3.3 Test T_3

Test T_3 is a Not-All-Equal test of long code B. While test T_1 and T_2 checks the consistency between tables A and B, T_3 is a test for a single table B. The procedure of the test T_3 is following.

1. The verifier chooses a random set $W \in \mathcal{W}$. It expects $B : \mathcal{F}_{\mathcal{M}_W} \mapsto \{-1,1\}$ as a long code of an assignment for W. The verifier picks three functions $g_1, g_2, g_3 : \mathcal{M}_W \mapsto \{-1,1\}$ as following.

$$(g_1(y), g_2(y), g_3(y)) = \begin{cases} (-1,1,1),(1,-1,1),(1,1,-1) & \\ \qquad\qquad \text{with each prob. } \frac{2}{3}-p & \\ (1,-1,-1),(-1,1,-1),(-1,-1,1) & \\ \qquad\qquad \text{with each prob. } p-\frac{1}{3} & \end{cases}$$

2. The verifier accepts iff Not-All-Equal$(B(g_1), B(g_2), B(g_3))$.

Obviously, this test always accepts proper long code, thus T_3 has perfect completeness. According to Fourier analysis, acceptance probability for NO instance is denoted as

$$\Pr[\mathrm{Acc}] \leq E_W[1-\hat{B}_\emptyset^2]+O(\epsilon).$$

3.3.4 Test T_4

Test T_4 is also a test for long code B. The procedure of the test T_4 is following.

1. The verifier chooses a random set $W \in \mathcal{W}$. It expects $B : \mathcal{F}_{\mathcal{M}_W} \mapsto \{-1,1\}$ as a long code of an assignment for W. The verifier picks three functions $g_1, g_2, g_3 : \mathcal{M}_W \mapsto \{-1,1\}$ as following.

$$(g_1(y), g_2(y), g_3(y)) = \begin{cases} (-1,-1,1) & \text{with prob. } 2p-1 \\ (-1,1,1),(1,-1,1) & \text{with each prob. } 1-\frac{3p}{2} \\ (-1,1,-1),(1,-1,-1) & \text{with each prob. } \frac{p}{2} \end{cases}$$

2. The verifier accepts iff $(B(g_1), B(g_2), B(g_3)) \neq (-1,-1,-1)$.

This test always accepts a proper long code, thus T_4 has perfect completeness. Dividing acceptance condition in some cases and using Fourier analysis, the acceptance probability of this test for NO instance can be bounded with the following expression.

$$\Pr[\mathrm{Acc}] \leq E_W\left[1+\frac{\hat{B}_\emptyset-\hat{B}_\emptyset^2}{2}\right]+O(\epsilon).$$

3.4 The PCP Construction

We construct the PCP combining four tests described in the previous section. Let $\eta \geq 0$ be a parameter determined later. The verifier performs each test with the following probability distribution.

$$\Pr[T_1] = \Pr[T_2] = \frac{4\eta+4}{12+9\eta},$$
$$\Pr[T_3] = \frac{\eta}{12+9\eta}, \Pr[T_4] = \frac{4}{12+9\eta} \qquad (1)$$

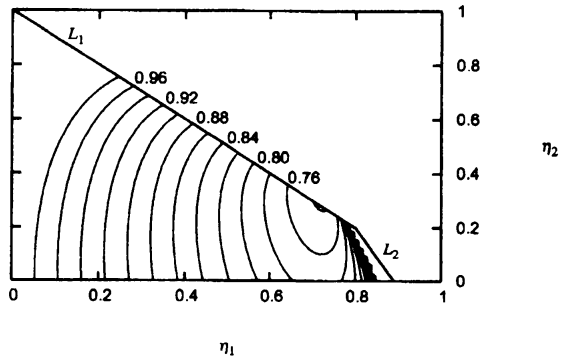

Figure 1. Change in acceptance probability according to execution probability of the tests

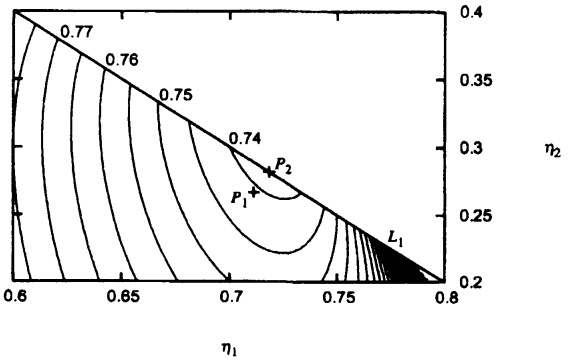

Figure 2. Enlaged view around the minimum value

And thus the acceptance probability for NO instance is

$$\begin{aligned}
&\Pr[\mathrm{Acc}] \\
&\le E_{W,U}\left[\frac{4\eta+4}{12+9\eta}\frac{5-\hat{A}_\emptyset-2\hat{B}_\emptyset+2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2+3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8}\right. \\
&\quad+\frac{4\eta+4}{12+9\eta}\frac{5+\hat{A}_\emptyset-2\hat{B}_\emptyset-2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2-3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8} \\
&\quad\left.+\frac{\eta}{12+9\eta}(1-\hat{B}_\emptyset^2)+\frac{4}{12+9\eta}\left(1+\frac{\hat{B}_\emptyset}{2}-\frac{\hat{B}_\emptyset^2}{2}\right)\right]+O(\epsilon) \\
&=\frac{6\eta+9+\eta^2-E_{W,U}[(\hat{B}_\emptyset+\eta)^2]}{12+9\eta}+O(\epsilon) \\
&\le\frac{6\eta+9+\eta^2}{12+9\eta}+O(\epsilon).
\end{aligned}$$

Above probability is minimized when $\eta = \frac{1}{3}$. Now the probability of each test becomes

$$\Pr[T_1]=\Pr[T_2]=\frac{16}{45},\Pr[T_3]=\frac{1}{45},\Pr[T_4]=\frac{12}{45}$$

and finally the soundness turns out to be

$$\Pr[\mathrm{Acc}]\le\frac{20}{27}+O(\epsilon)\simeq 0.74074.$$

All the tests have perfect completeness and whole soundness is shown as above, that the proof of theorem 6 is completed.

4 Improving the Soundness

We can see that the analysis of section 3 is not tight. In this section, we show that the soundness can be made smaller under the same conditions. As a corollary, we have the inapproximability results for a constraint satisfaction problem where each constraint depends on at most three variables (3-CSP).

4.1 Optimizing the Probability of the Tests

In Khot and Saket's analysis, they determine the probability of selecting one of the four tests based on only one parameter η, although the degree of freedom is bounded by 3 due to constraint $\Pr[T_1]+\Pr[T_2]+\Pr[T_3]+\Pr[T_4]=1$, or 2 if we force $\Pr[T_1]=\Pr[T_2]$ to eliminate $\hat{A}_\emptyset$. Furthermore, they did not explain why the probability of each test is denoted as equation (1). So now we set the distribution as $\Pr[T_1]=\Pr[T_2]=\frac{\eta_1}{2},\Pr[T_3]=1-\eta_1-\eta_2,\Pr[T_4]=\eta_2$ and calculate the acceptance probability for NO instance ψ.

$$\begin{aligned}
&\Pr[\mathrm{Acc}] \\
&\le E_{W,U}\left[\frac{\eta_1}{2}\frac{5-\hat{A}_\emptyset-2\hat{B}_\emptyset+2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2+3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8}\right. \\
&\quad+\frac{\eta_1}{2}\frac{5+\hat{A}_\emptyset-2\hat{B}_\emptyset-2\hat{A}_\emptyset\hat{B}_\emptyset+\hat{B}_\emptyset^2-3\hat{A}_\emptyset\hat{B}_\emptyset^2}{8} \\
&\quad\left.+(1-\eta_1-\eta_2)(1-\hat{B}_\emptyset^2)+\eta_2\left(1+\frac{\hat{B}_\emptyset}{2}-\frac{\hat{B}_\emptyset^2}{2}\right)\right]+O(\epsilon) \\
&=1-\frac{1}{8}\frac{-28\eta_1^2-8\eta_1\eta_2-4\eta_2^2+24\eta_1}{8-9\eta_1-4\eta_2} \\
&\quad-\frac{8-9\eta_1-4\eta_2}{8}E_{W,U}\left[\left(\hat{B}_\emptyset-\frac{2\eta_2-\eta_1}{8-9\eta_1-4\eta_2}\right)^2\right] \\
&\quad+O(\epsilon) \qquad (2) \\
&\le 1-\frac{1}{8}\frac{-28\eta_1^2-8\eta_1\eta_2-4\eta_2^2+24\eta_1}{8-9\eta_1-4\eta_2}+O(\epsilon) \qquad (3)
\end{aligned}$$

Here, since the sum of the probability of selecting each test is 1, η_1 and η_2 must satisfy the constraint

$$L_1:\eta_1+\eta_2\le 1.$$

The third term in the equation (2) contains $\hat{B}_\emptyset$, whose value may vary according to the input of the PCP. To bound the acceptance probability for every input, the term must have non-positive value. It means that η_1 and η_2 must satisfy the

following constraint.

$$L_2 : 8 - 9\eta_1 - 4\eta_2 > 0$$

Because η_1 and η_2 represent probability, there are also constraints $\eta_1, \eta_2 \geq 0$. We would like to find the values of η_1 and η_2 which minimizes the acceptance probability under these conditions.

The acceptance probability of the whole test is shown in the figure 1, where η_1 and η_2 vary satisfying $\eta_1, \eta_2 \geq 0, \eta_1 + \eta_2 \leq 1, 8 - 9\eta_1 - 4\eta_2 > 0$ In the plot area, bold line represents the border of the constraints L_1 and L_2, and the numbers in the area represents the value of the level lines. The figure shows that inside the area where the constraints are satisfied, obviously there is no point where the acceptance probability Pr[Acc] takes local minimum value, so Pr[Acc] takes minimum on the boundary of constraint L_1. In the PCP construction of Khot and Saket, the probability of each test is defined as the point

$$P_1 : (\eta_1, \eta_2) = \left(\frac{32}{45}, \frac{12}{45}\right),$$

shown in the figure 2. The figure shows us that the acceptance probability at the point is clearly apart from minimum value.

To obtain the minimum value of Pr[Acc] on the boundary of the constraint L_1, we set $\eta_2 = 1 - \eta_1$. Substituting $\eta_2 = 1 - \eta_1$ in the equation (3), we have

$$\begin{aligned}
&\Pr[Acc] \\
&\leq 1 - \frac{1}{8}\frac{-28\eta_1^2 - 8\eta_1(1-\eta_1) - 4(1-\eta_1)^2 + 24\eta_1}{8 - 9\eta_1 - 4(1-\eta_1)} + O(\epsilon) \\
&= 1 - \frac{1}{2}\frac{-6\eta_1^2 + 6\eta_1 - 1}{4 - 5\eta_1} + O(\epsilon).
\end{aligned}$$

Differentiating the above expression with respect to η_1 and letting its derivative $= 0$, we have

$$\begin{aligned}
\frac{d}{d\eta_1}\left[1 - \frac{1}{2}\frac{-6\eta_1^2 + 6\eta_1 - 1}{4 - 5\eta_1} + O(\epsilon)\right] &= 0 \\
-\frac{1}{2}\frac{30\eta_1^2 - 48\eta_1 + 19}{4 - 5\eta_1} &= 0 \\
30\eta_1^2 - 48\eta_1 + 19 &= 0.
\end{aligned}$$

Solving the equation with respect to η_1, we have

$$\eta_1 = \frac{24 \pm \sqrt{6}}{30},$$

where $\eta_1 = \frac{24+\sqrt{6}}{30}$ violates the constraint $L_2 : 8 - 9\eta_1 - 4\eta_2 > 0$, so double sign is determined by $-$. Above calculation shows that the value of η_1 and η_2 where Pr[Acc] takes minimum in the area satisfying all constraints are

$$P_2 : (\eta_1, \eta_2) = \left(\frac{24 - \sqrt{6}}{30}, \frac{6 + \sqrt{6}}{30}\right).$$

The point P_2 is shown in the figure 2. Now the probabilities of these tests are

$$\begin{aligned}
\Pr[T_1] = \Pr[T_2] &= \frac{24 - \sqrt{6}}{60} \simeq 0.35918, \\
\Pr[T_3] = 0, \Pr[T_4] &= \frac{6 + \sqrt{6}}{30} \simeq 0.28165,
\end{aligned}$$

which implies that the test T_3 is unnecessary. Finally, the acceptance probability of the PCP when an input 3-SAT-5 formula ψ is NO instance, that is, the soundness becomes

$$\Pr[\text{Acc}] \leq \frac{16 + \sqrt{6}}{25} + O(\epsilon) \simeq 0.73798 \leq \frac{20}{27} \simeq 0.74074,$$

so we have succeeded to improve the result of Khot and Saket. Now the theorem 6 is improved as the following.

Theorem 10.

$$\forall \epsilon > 0, \quad \text{NP} = \text{naPCP}_{1, \frac{16+\sqrt{6}}{25}+\epsilon}[O(\log n), 3]$$

Furthermore, we simplified the PCP by showing that the test T_3 is not necessary. We denote this PCP by $\text{PCP}_{\{T_1, T_2, T_4\}}$.

4.2 Inapproximability of 3 CSP

Constraint Satisfaction Problem (CSP) is a problem to obtain a kind of object which satisfies the given constraints. If the constraints are given by a form of Boolean functions depending on the set of some variables, the object corresponds to an assignment to variables. Here, we discuss the inapproximability of CSPs where each constraint depends on up to three Boolean variables. Such CSPs are called 3-CSPs.

We assume that each constraint in CSPs has weight. The weights are normalized so that their sum equals 1. Our objective in this 3-CSPs is to obtain an assignment that maximizes the total weights of satisfied constraints. A value of the 3-CSP instance G, which is denoted as $\omega(G)$, is defined as a maximum value of the total weights of satisfied constraints. The instance of $\omega(G) = c$ is called c-satisfiable and 1-satisfiable instances are simply called satisfiable.

We construct 3-CSP instances from our 3-bit PCP. Each tests T_1, T_2 and T_4 makes randomly chosen three-bit queries to the proof. We replace each query bit by a Boolean variable and replace the acceptance condition by a constraint. The weight of each constraint is the probability with which the triplet of queries is selected in each test. In this way, the tests are converted to 3-CSP instances G_{T_1}, G_{T_2} and G_{T_4} respectively, and we construct a new 3-CSP instance G_{T_1,T_2,T_4} by combining these 3-CSP instances. The weight of each constraint in G_{T_1,T_2,T_4} is defined as the weight in its original instance G_{T_1}, G_{T_2} or G_{T_4} multiplied by the probability of the corresponding test in $\text{PCP}_{\{T_1,T_2,T_4\}}$.

In the above constructed 3-CSP instance G_{T_1,T_2,T_4}, the value $\omega(G_{T_1,T_2,T_4})$ is equal to the acceptance probability of $\text{PCP}_{\{T_1,T_2,T_4\}}$ for the input 3-SAT-5 instance ψ. This is

obvious due to the conversion from $\mathrm{PCP}_{\{T_1,T_2,T_4\}}$ to the 3-CSP instance G_{T_1,T_2,T_4}. Hence when ψ is an YES instance, the value of $\omega(G_{T_1,T_2,T_4})$ corresponds to the completeness of $\mathrm{PCP}_{\{T_1,T_2,T_4\}}$, and when ψ is NO instance, the value of $\omega(G_{T_1,T_2,T_4})$ corresponds to the soundness. From these facts, we show the following inapproximability.

Theorem 11. *For any $\epsilon > 0$, it is NP-hard to distinguish between a satisfiable 3-CSP instance and a $(\frac{16+\sqrt{6}}{25} + \epsilon)$-satisfiable 3-CSP instance.*

Proof. In 3-CSP G_{T_1,T_2,T_4}, the sum of the weights of satisfied constraints equals to the acceptance probability of $\mathrm{PCP}_{\{T_1,T_2,T_4\}}$. Suppose that there is a polynomial time approximation algorithm which distinguish satisfiable 3-CSP instance from $(\frac{16+\sqrt{6}}{25}+\epsilon)$-satisfiable 3-CSP instance. Then the algorithm can determine whether the acceptance probability of $\mathrm{PCP}_{\{T_1,T_2,T_4\}}$ for some 3-SAT-5 instance ψ is higher than the soundness $\frac{16+\sqrt{6}}{25} + \epsilon$ or not. This implies that the algorithm can determine whether the 3-SAT-5 instance ψ is satisfiable or not. Having such an algorithm, we can solve an arbitrary NP decision problem by reducing the problem to a 3-SAT-5 instance ψ, that means the above mentioned task is NP-hard. □

5 Conclusion

In this paper, we improved the soundness of a non-adaptive 3-bit PCP from the previous $\frac{20}{27} + \epsilon$ to $\frac{16+\sqrt{6}}{25} + \epsilon$. We have also shown that the number of the tests used in the PCP can be decreased from 4 to 3. This results in a simplification of the PCP construction. As a corollary, we showed an improved inapproximability results of 3-CSPs.

On the algorithmic side, Zwick showed $\frac{5}{8}$ approximation ratio algorithm for any satisfiable 3-CSP instances. This algorithm was proved to be optimal by O'Donnel and Wu [11] under Khot's d-to-1 Conjecture. That is, they constructed a non-adaptive 3-bit PCP with soundness $\frac{5}{8} + \epsilon$ and perfect completeness under the conjecture. However d-to-1 Conjecture still remains open (similar to the Unique Games Conjecture) , so obvious future work is decreasing the soundness without such assumptions.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM*, 45(1):70–122, 1998.

[3] M. Bellare, O. Goldreich, and M. Sudan. Free bits, pcps and nonapproximability—towards tight results. *SIAM Journal of Computing*, 27(3):804–915, 1998.

[4] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):27–36, 1998.

[5] V. Guruswami, D. Lewin, M. Sudan, and L. Trevisan. A tight characterization of np with 3 query pcps. In *Proc. of 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 8–17, 1998.

[6] J. Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[7] S. Khot. Hardness of coloring 3-colorable 3-uniform hypergraphs. In *Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 23–32, 2002.

[8] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. of the 34th ACM Symposium on the Theory of Computing*, pages 767–775, 2002.

[9] S. Khot. New techniques for probabilistically checkable proofs and inapproximability results. In *PhD thesis, Princeton University*, Princeton, New Jersey, USA, 2003.

[10] S. Khot and R. Saket. A 3-query non-adaptive pcp with perfect completeness. In *Proc. of the 21st Annual IEEE Conference on Computational Complexity*, pages 159–169, 2006.

[11] R. O'Donnell and Y. Wu. Conditional hardness for satisfiable 3-csps. *Submitted*, Available at http://www.cs.cmu.edu/ odonnell/papers/3bit-hardness.pdf.

[12] R. Raz. A parallel repetition theorem. *SIAM Journal of Computing*, 27(3):763–803, 1998.

[13] U. Zwick. Approximation algorithms for constraint satisfaction problems involving at most three variables per constraint. In *Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210, 1998.